

PATENT

Attorney Docket No: CSCO-007/92821

**TRACING LAYER-2 ROUTE IN NETWORKS BASED
ON BROADCAST MEDIUM**

Inventor(s)

Sailaja S K ATTILI Bangalore, India Citizenship: India	Sridhar ASWATHNARAYAN San Jose, CA 95135 Citizenship: Indian
--	--

Assignee:

Cisco Technology, Inc.
A California Corporation
170 W. Tasman Drive
San Jose, CA 95134
Telephone: (408) 525-9706
Fax: (408) 526-5952

Attorney:

Law Firm of Naren Thappeta
Customer Number: 26392
www.iphorizons.com

TRACING LAYER-2 ROUTE IN NETWORKS BASED ON BROADCAST MEDIUM

Background of the Invention

Field of the Invention

5 The present invention relates to communication networks, and more specifically to a method and apparatus for tracing routes in networks based on broadcast medium (e.g., Ethernet).

Related Art

A networking environment generally includes network devices connecting user systems. User systems are used to implement user applications, which provide corresponding features to the users using the end systems. A network device ("hereafter device") generally refers to a device which forwards data packets ("packets") between end systems to support various network applications.

10 It often helpful to know the specific intermediate devices present in a path taken by packets from a source system (e.g., computer system) to a destination system. For example, when troubleshooting a perceived problem of low data transfer throughput between the source and destination systems, a network administrator may wish to know the intermediate devices so that the problem can potentially be isolated to one of more of the intermediate devices. The path formed by the sequence of the intermediate devices is often referred to as a route between the corresponding source and destination systems.

15 Utilities which allow the determination of a route (between a source and a destination system) are commonly referred to as trace-route utilities. The well-known "Trace-Route" software program

which provides the layer-3 devices between two systems is an example of such an utility. Such an utility generally lists the routers (which operate at the networking layer, also referred as layer-3) present in the path between a source and destination system. Further details of the Trace-Route utility are described in further detail in books entitled, "Internetworking with TCP/IP Volume 1", by Douglas E. Comer, and "TCP/IP Illustrated Volume 1", by W. Richard Stevens, which are both incorporated in their entirety herewith.

Unfortunately, the Trace utility operating at layer-3 level typically does not provide information on any layer-2 devices present in the path between a source and a destination system. The feature is particularly problematic in networks based on broadcast medium since the route taken by packets is not generally pre-configured (i.e., not pre-provisioned, for example, as in ATM networks) and the route is generally determined dynamically. A broadcast medium generally refers to a medium in which several devices would generally be capable of receiving a packet intended for even a point-to-point communication. An example environment may be based on Ethernet/802.3 technology as is well known in the relevant arts.

With respect to routes, in Ethernet/802.3 environment well known in the relevant arts, each device determines the specific direction (usually port on the device) a system is in, by examining a source address present in a received packet header. The path may vary if redundancy is present in the network. In addition, a network administrator may not be able to rule out the introduction of additional devices in the network by other persons. Accordingly, a network administrator having responsibility for a network based on broadcast medium may have particular motivation to know the presence of any intermediate layer-2 devices between two end systems.

Thus, what is needed is a method and apparatus which traces the layer-2 route between a source system and destination system.

Summary of the Invention

5 The present invention enables a network administrator to enter a command and determine a list of any intermediate layer-2 devices between a source system and a destination system in a network implemented using broadcast medium. The administrator may enter the command using a command line interface (CLI) on a device ("receive device"), and the receiving device may process the command as described below.

10 The receiving device first determines a first layer-2 device which is connected directly to the source system. If the destination system is not connected to the receiving device, the receiving device may execute a sequence of loops to determine each intermediate device in the route between the source and destination system. For the purpose of implementing the loops, the receiving device may logically view the first layer-2 device as a present layer-2 device if the destination system is also not directly connected to the first layer-2 device.

15 The receiving device may send a request packet to the present layer-2 device requesting information on whether the destination system is connected directly to the system. A system is deemed to be directly connected to a device if there are no intermediate devices (and layer-3 devices) between the system and the device. In the described embodiment, the direct connection is present if the system is directly connected to a port of the layer-2 device.

The receiving device may then receive a response packet from the present layer-2 device. The response packet indicates whether the destination system is connected directly to the present layer-2 device, and if not, a subsequent layer-2 device in a route to the destination system if the destination system is not connected directly to the present layer-2 device. The subsequent layer-2 device represents the next (after the present layer-2 device) layer-2 device in the route to the destination system.

The above described sequence of sending and receiving are repeated by using the subsequent layer-2 device in the place of the present layer-2 device until the response packet indicates that the destination system is directly connected to the presently layer-2 device. After one or more iterations (instances) of the sequence, a response packet indicates that the destination system is connected directly to the corresponding present layer-2 device. Thus, the sequence of layer-2 devices detected in the iterations forms the route from the source device to the destination device. The route is said to be traced.

According to another aspect of the present invention, the route between a source system and a destination system is determined even when the receiving device is not directly connected to the source device. To provide such a feature, the receiving device first locates a directly connected device which is connected directly to the source system. In one embodiment, the directly connected device is determined by tracing the route from the receiving device to the first system.

Alternatively, a multicast based approach in which the receiving device broadcasts a packet, and the receiving devices are designed to respond if the source system is connected directly. The

directly connected device may then be used as the present layer-2 device, and a request packet may be sent to the present layer-2 device. The sequence of sending and receiving packets is continued until the desired route is entirely trace.

One more aspect of the present invention enables a network administrator to issue the route request commands by using a command line interface on a receiving device. Alternatively, the features may be embodied in general purpose computer systems which generate request packets and process response packets as described above. An embodiment is implemented substantially in the form of software. The request and response packets may be implemented using well-known protocols such as UDP/IP.

Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

Brief Description of the Drawings

The present invention will be described with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram illustrating an example communication environment in which the present invention can be implemented;

Figure 2 is a flow chart illustrating a method in accordance with the present invention;

Figure 3 is a flow chart illustrating a method using which the route between a source system and

a destination system may be determined even when the receiving device is not directly connected to the source system;

Figure 4 is a block diagram illustrating the internals of a network device in an embodiment of the present invention; and

Figure 5 is a block diagram illustrating an embodiment of network device implemented substantially in the form of software.

Detailed Description of the Preferred Embodiments

1. Overview and Discussion of the Invention

The present invention enables a network administrator to check the layer-2 route between a source system and a destination system present in a network. According to another aspect of the present invention, the device receiving the command determines the route by interfacing with each intermediate device in the route, and provides the information to the network manager in a suitable format. The manner in which the device makes such a determination is described below.

The invention is described below with reference to an example environment for illustration. It should be understood that numerous specific details, relationships, and methods are set forth to provide a full understanding of the invention. One skilled in the relevant art, however, will readily recognize that the invention can be practiced without one or more of the specific details, or with other methods, etc. In other instances, well-known structures or operations are not shown in detail to avoid obscuring the invention. Furthermore the invention can be implemented in several other environments.

2. Example Environment

Figure 1 is a block diagram illustrating an example network environment 100 in which the present invention can be implemented. Network environment 100 is shown containing layer-2 devices 110, 120, 130 and 150 forming a local area network (LAN) 190. Devices 110, 120, 130, and 150 provide connectivity to systems (101 and 102), (121 and 122), (131) and (151) respectively. The operation of all components is described below in further detail.

For illustration, it is assumed that LAN 190 is implemented using Ethernet (or IEEE 802.3 protocol) well known in the relevant arts. The protocols are described in IEEE Standard 802.3, 2000 Edition, which is incorporated in its entirety herewith. In general, LAN 190 can be implemented using any broadcast medium. Each of the paths 112, 123, 125 and 135 are shown connecting respective pairs of devices implemented using broadcast protocols (Ethernet in the present embodiment).

Devices 110, 120, 130 and 150 correspond to layer-2 devices operating consistent with Ethernet protocol. In one embodiment, each device corresponds to Catalyst 5000/ 6000 series products (modified in accordance with several aspects of the present invention) available from Cisco Systems, Inc. Each device contains a port (not shown) which provides the physical interface to connect to the corresponding one of paths or one of the systems. Thus, device 110 may contain four ports connecting respectively to systems 101-103 and path 112.

A network administrator may enter a command on any of the devices implemented consistent with various aspects of the present invention to determine any intermediate devices present between two user systems. Many conventional layer-2 devices may be modified to provide such a feature to

the network administrators. The manner in which a device may provide the feature is described below first with reference to a method (Figure 2) and then with an example architecture (Figure 4).

3. Method

Figure 2 is a flow-chart illustrating a method according to which a device (described with reference to device 130 for conciseness) may determine the route to a destination system. The method begins in step 201, in which control immediately passes to step 210.

In step 210, device 130 ("receiving device") receives a trace command requesting information on any layer-2 devices present between a source device and a destination device. The devices may be specified in various ways, for example, as layer-2 address (e.g., Ethernet MAC address), layer-3 address (e.g., IP address) or a name (e.g., machine name which can be resolved into a layer-3 address). For present, it is assumed that the source system is connected directly to a port of the receiving device.

In step 220, the receiving device determines the layer-2 addresses corresponding to the two systems. If the systems were specified using layer-2 addresses, the determination is typically complete upon mere examination of the corresponding data. Even in the case a system is specified using other approaches, the layer-2 address may be determined in a known way.

In step 230, the receiving device determines a port on which communication can be established with the destination device. The determination depends on the technology used to implement LAN 190. An example embodiment with reference to Ethernet environment is described below in detail.

In step 240, the receiving device checks whether the destination system is connected directly to the receiving device. A system may be determined to be connected to a device directly if there are no other devices in between the system and device. With reference to Figure 1, a system would be deemed to be connected directly to a device, if the system is connected to a port of the device.

5 If the destination system is determined to be connected to the port of the device, the entire route information to the destination system is deemed to be determined, and the method may end after providing the information to the network administrator. If the destination system is determined not to be connected to the device, then control passes to step 270.

In step 270, the receiving device sends a request packet to an intermediate device connecting to the port to indicate where the destination system is. The intermediate device generates a response packet indicating whether the destination system is connected directly to a port connected to the intermediate device or a next intermediate device which is in the path to the destination system.

15 In step 280, the receiving device receives the packet from the intermediate device and control then passes to step 240. If the packet indicates that the destination system is connected directly to the port (of the intermediate device), control passes to step 299 as the entire path to the destination system has been detected. Otherwise, control passes to step 270, which is performed treating the next intermediate device as "the intermediate device".

The loop of steps 240, 270 and 280 is repeated until the entire path is determined. Thus, the method of Figure 2 can be used to determine the path from a source system (connected directly to a

receiving device) to a destination system as described above. The same approach can be used to determine the path between the two devices also. The description is continued with respect to some general considerations in example environment(s) for determining the next neighbors.

4. Determining Next Neighbors in the Route in an Example Environment

5 In one embodiment, LAN 190 is implemented using Ethernet protocol/technology and systems 101-103, 121, 122, 131 and 151 are all implemented consistent with the Ethernet protocol. It is further assumed that each of the systems and devices has a unique Internet Protocol (IP) address.

In one such embodiment, the determination of devices in the route to a destination system may entail three tasks: (1) determining in each device in the path the port which provides communication to the destination system; (2) determining whether the port is connected to the destination system or whether there is another intermediate device connected to the port; and (3) sending a request packet to the next intermediate device if the port is not connected to the destination system.

With respect to (1), assuming the received command contains an IP address for the destination system, device 130 may translate the IP address into an Ethernet address based on local table lookup.
15 If the Ethernet Address is not available, a protocol such as Address Resolution Protocol (ARP) may be used to determine the Ethernet Address. Typically, a packet with the Ethernet Address of the source system is received in response, and the port on which the packet is deemed to provide the communication with the destination system.

With respect to step (2), device 130 may determine that device 150 is the adjacent device on the other side of path 135 by using protocols such as Cisco Discovery Protocol (CDP). In general, a device implemented according to CDP sends information to neighbors indicating the fact that it (i.e., the device) is a neighbor on that port, and IP address which can be used to contact the device. Thus, for example, device 130 may determine that device 150 is the neighbor on path 135 using CDP related packets received from device 150, and the IP address of device 150.

Device 130 may determine that the port is directly connected to the destination system if no CDP related messages are not received on the port. However, other protocols and approaches may be used in determining whether the destination system is connected to the port of a device and the manner in which to communicate with the next neighbor in the route to the destination system, as will be apparent to one skilled in the relevant arts based on the disclosure provided herein. Such other approaches and protocols are contemplated to be within the scope and spirit of several aspects of the present invention.

Thus, once the next device and the IP addresses of the next device in the route are determined, a request packet may be generated to the next device requesting whether the destination system is connected directly to a port of the next device or to provide information on a subsequent device in the route to the destination system. If the destination system is not connected to a port of the next device, an additional request packet may be sent to the subsequent device. The sequence of request and response packets may be used until a device indicates that the destination system is connected to a port of the device.

Thus, the above described method(s) and approach(es) can be used to determine the route to a destination device when the source device is connected directly to a port of the receiving device. An aspect of the present invention allows the route to be determined even in situations when a receiving device is not connected directly to the source system as described below.

5

5. Situation When a Receiving Device is Not Connected to Source System

Figure 3 is a flow chart illustrating a method using which the route from a source system to a destination system can be determined even when the source system is not directly connected to a receiving device. The method is described with respect to a situation in which device 130 ("receiving device") receives a command to provide route information from system 101 ("source system") to system 151 (destination system).

The method begins in step 301, in which control immediately passes to step 310. In step 310, device 130 receives the trace command. In step 330, the receiving device determines that the source system is not connected to any of the receiving device's ports, for example, by first determining a port providing communication to the source system and then using the absence of CDP related information on the determined port as described above. Other approaches may as well be used depending on the available products and technologies.

In step 350, the device ("connecting device") that connects directly to the source system may be located treating the source system as a destination system ("temporary destination system"), and tracing the route from the receiving device to the source system. With respect to the present example, device 130 may first determine that port connecting to path 135 needs to be used for communicating

with system 101. It is here assumed that a protocol such as spanning tree protocol has earlier determined that paths (135, 125 and 112) should be used instead of paths (123 and 112).

Assuming a protocol such as CDP (Cisco Discovery Protocol) provides information that device 150 is connected on the other side of path 135, device 150 may send a request packet including information on the temporary destination system. Device 150 indicates that device 120 (identified by a corresponding ID address) should be contacted next to continue tracing the route to the temporary destination.

The sequence of request and responses may be continued until device 110 indicates that the temporary destination system 101 is connected to a port of device 110. Thus, by tracing the route to the source system specified in a command, a receiving device may locate the connecting device (device 110 in the present example) connecting to the source system.

An alternative embodiment may be implemented using a multicast (including broadcasts) packet, with the multicast packet containing the information of the temporary destination system (or the source system specified in the command). Each device may be designed to respond if the temporary destination system is connected directly to the device. Many implementations consistent with such an approach will be apparent to one skilled in the relevant arts based on the disclosure herein, and such implementations are contemplated to be within the scope and spirit of the present invention.

In step 370, the source device (130) may trace the path from the device (110) connecting to the source system. The method of Figure 2 described above may be used to determine the route.

Thus, an aspect of the present invention enables a network administrator to enter a trace command on any device of network 190 irrespective of where the source and destination systems are present. The description is continued with reference to an example packet format which can be used for the request and response packets between devices.

6. Packet Format

In one embodiment, UDP/IP packets with a similar format is used for both requests and responses. Only the details of UDP and IP as relevant to an understanding of the presented specific examples are described herein. For further details of UDP and IP, the reader is referred to documents entitled RFCs 791 and 768, available from www.ietf.org, which are both incorporated in their entirety herewith. The packet format is described below in further detail.

Any port number not inconsistent without prior uses (e.g., the standard ports used for applications such as Telenet, FTP) may be used to indicate that the UDP packet relates to requests and responses, and the data portion of the UDP packet is to be interpreted according to the further details as noted below:

Byte 1 (packet type): 1 = trace destination system; 2 = trace source system (i.e., when the source system is not connected to the receiving device); 3 = reply for destination trace; and 4 = reply for source trace.

Byte 2 (version Number): value indicates version number of the implementation.

Bytes 3-4: length of the packet.

Byte 5: number of attributes.

Bytes 6-end (Attributes): The requested or provided information according to a pre-determined convention. The first byte of an attribute may specify the attribute type, second byte the length of the attribute, and the remaining bytes the data necessary for the attribute.

Examples of attributes are the IP address of the system being traced, the IP (or MAC/Ethernet) address of the next device, the port to which the system is connected, the etc. The attributes may be extended to provide other features (e.g., port speed, port MAC address) which may be of interest to a network administrator.

An example request packet may contain MAC address of the source system, MAC address of the destination system, and IP address of the receiving device. The MAC address of the destination system may be used to determine the next device in the path and the IP address of the receiving device may be used to send a response packet. An example response packet may contain the IP address of the device sending the response packet, the IP address of the next device in the path to the destination system, and port related information (e.g., speed of the ports).

Using the packet format of above, a receiving device (i.e., device receiving a trace command) may trace the route from any source system to a destination system.

7. Network Device

Figure 4 is a block diagram illustrating the details of device 130 in one embodiment. Device 130 is shown containing inbound interface 410, parser 420, response processor 430, user interface 440, MAC (medium access control) lookup 450, MAC table 455, next hop block 460, next hop table

465, generate request block 470, tables update block 480 and outbound interface 490. Each component is described below in detail.

Each component of device 130 may be implemented in a combination of one or more of hardware, software and firmware. In general, when throughput performance is of primary consideration, the implementation is performed more in hardware (e.g., in the form of an application specific integrated circuit). When cost is of primary consideration, the implementation is performed more in software (e.g., using a processor executing instructions provided in software/firmware). Cost and performance can be balanced by implementing device 130 with a desired mix of hardware, software and/or firmware.

Inbound interface 410 is shown receiving packets from three ports (shown as logically as circles) corresponding to paths 123, 135 and 133 of Figure 1, and forwards the packets to parser 420. Inbound interface 410 provides the electrical and other protocol interfaces necessary to receive packets from various paths, and may be implemented in a known way. Outbound interface 490 is also described similarly, except that the packets received from generate request 470 are transmitted in the outbound direction on the same three ports.

Parser 420 forwards the packets (originating from systems and other devices) received from inbound interface 410 to one of response processor 430, MAC lookup block 450 and tables update block 480. The specific one of the units to which to forward each packet is determined typically by examining the content of the packet.

Response processor 430 receives a response packet generated by another device. The response packet is generally generated in response to a request packet sent by generate request/response block 470. The response processor indicates whether the responding device is connected to the destination system or a next device in the route to the destination device.

5 If the destination device is connected to the responding device, then the route to the destination device (or temporary destination device) is entirely traced. If a response packet contains information for a next device in the route, response processor 430 interfaces with generate request/response processor 470 to generate a request directed to the next device.

User interface 440 provides a convenient interface to a network administrator. Thus, user interface 440 may receive a command from the user, and generate the route information. In an embodiment, user interface 440 updates a display as each device in the route is determined (by examining a response packet). With respect to input interface, a network administrator may enter a command using a command line interface (CLI) provided with device 130.

User interface 440 may interface with MAC lookup 450 to initiate the processing of a command. User interface 440 may also contain the logic to determine and process commands which specify a source system not connected to device 130. The logic first traces the route to the source system to determine a device connecting directly to the source system, and then starts tracing the route from the connecting device to the destination system as described above in further detail. The display to the network administrator may indicate that the source is being traced first, and then each intermediate device in the route from the source system to the destination.

MAC table 455 may contain entries mapping each IP address to a corresponding MAC (e.g., Ethernet) address. MAC table 455 entries may be populated in a known way. Next hop table 465 contains information on a device connecting on the other end of the each path connecting to a port. In one embodiment, next hop table 465 is populated by protocols such as CDP, in which neighbor devices broadcast information as described in sections above. MAC table 455 and next hop table 465 may be implemented in a Random Access Memory, to attain a high throughput performance.

Tables update block 480 updates the entries in MAC table 455 and next HOP table 465 based on the packets received from various sources. The implementation of tables update block 480 depends on the protocols and technologies being used in LAN 190. Tables update block 480 may be implemented in a known way.

MAC lookup block 450 may receive a trace command (from user interface 440) and determine an Ethernet Address corresponding to any systems in the command. The Ethernet address(s) may be passed to next hop block 460. Next hop block 460 determines whether a system specified in a command is connected directly to a port on device 130. If the system is not connected to the port, next hop block 460 determines the address of the a neighbor (i.e., connected to a path provided through a port on device 130).

The combination of MAC lookup block 450 and next hop block 460 may also process a request packet (received from parser 420) generated by other devices (e.g., 110). In both cases (i.e., when processing a request and when processing a command), the information necessary for the next

device in the route is generated. An embodiment of next hop block 460 uses information provided by protocols such as CDP.

Generate request/response block 470 generates a request packet after response processor 430 processes a response packet indicating that the route needs to be traced further. The request packet is directed to a device specified in the response packet. A request packet may also be generate when initiating a trace. Generate request/response block 470 may generate a response packet when a request packet (generate by another device) is processed by next hop block 460.

Thus, the components described above may operate to generate requests and responses to trace the route between a source system and a destination system. Another embodiment of device 130 may be implemented substantially in software as described below in further detail. It should be noted that yet another embodiment of the present invention enables the embodiments to be implemented in computer systems (i.e., not operating as devices).

8. Software Implementation

Figure 5 is a block diagram illustrating the details of device 130 in one embodiment. Device 130 is shown containing processing unit 510, random access memory (RAM) 520, storage 530, output interface 560, network interface 580 and input interface 590. Each component is described in further detail below.

Output interface 560 provides output signals (e.g., display signals to a display unit, not shown) which can form the basis for a suitable user interface for a user to interact with Device 130. Input

interface 590 (e.g., interface with a key-board and/or mouse, not shown) enables a user to provide any necessary inputs to device 130. Output interface 560 and input interface 590 can be used to provide a command level interface and to display the route information to a network administrator.

Network interface 580 enables device 130 to send and receive various packets in accordance with the present invention. Network interface 580 may correspond to inbound interface 410 and/or outbound interface 490 of Figure 4. Network interface 580, output interface 560 and input interface 590 can be implemented in a known way.

RAM 530 and storage 530 may together be referred to as a memory. RAM 530 (can contain multiple memory units) may receive instructions and data on path 550 from storage 530. Secondary memory 530 may contain units such as hard drive 535 and removable storage drive 537. Secondary storage 530 may store the software instructions and data, which enable device 130 to provide several features in accordance with the present invention.

Some or all of the data and instructions may be provided on removable storage unit 540, and the data and instructions may be read and provided by removable storage drive 537 to processing unit 510. Floppy drive, magnetic tape drive, CD-ROM drive, DVD Drive, Flash memory, removable memory chip (PCMCIA Card, EPROM) are examples of such removable storage drive 537.

Processing unit 510 may contain one or more processors. Some of the processors can be general purpose processors which execute instructions provided from RAM 520. Some can be special purpose processors adapted for specific tasks (e.g., for memory/queue management). The special

purpose processors may also be provided instructions from RAM 520. In general processing unit 510 reads sequences of instructions from various types of memory medium (including RAM 520, storage 530 and removable storage unit 540), and executes the instructions to provide various features described above.

5 It should be understood that the software implementation of above can be implemented in other units such as general purpose computer systems (e.g., user system 101). In such a situation, the computer system may need to issue request packets and process response packets similar to device 130 described above. Thus, embodiments described above can be used to provide the route information between any two systems (devices). A network administrator may accordingly conveniently detect any intermediate devices between two systems.

9. Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

15